**What are some essential features of Spring Security?**

Some essential **features** of Spring Security include:

* Supports authentication and authorization in a flexible and comprehensive manner.
* Detection and prevention of attacks including session fixation, clickjacking, cross-site request forgery, etc.
* Integrate with Servlet API.
* Offers optional integration with Spring Web MVC (Model-View-Controller).
* Java Authentication and Authorization Service (JAAS) is used for authentication purposes.
* Allows Single Sign-On so that users can access multiple applications with just one account (username and password).

### What is Spring security authentication and authorization?

* **Authentication:**This refers to the process of verifying the identity of the user, using the credentials provided when accessing certain restricted resources. Two steps are involved in authenticating a user, namely identification and verification. An example is logging into a website with a username and a password. This is like answering the question Who are you?
* **Authorization:**It is the ability to determine a user's authority to perform an action or to view data, assuming they have successfully logged in. This ensures that users can only access the parts of a resource that they are authorized to access. It could be thought of as an answer to the question Can a user do/read this?

**What do you mean by basic authentication?**

RESTful web services can be authenticated in many ways, but the most basic one is basic authentication. For basic authentication, we send a username and password using the HTTP [Authorization] header to enable us to access the resource. Usernames and passwords are encoded using base64 encoding (not encryption) in Basic Authentication. The encoding is not secure since it can be easily decoded.

**Syntax:**

Value = username:password

Encoded Value = base64(Value)

Authorization Value = Basic <Encoded Value>

//Example: Authorization: Basic VGVzdFVzZXI6dGVzdDEyMw==

//Decode it'll give back the original username:password UserName:user123

### What do you mean by session management in Spring Security?

As far as security is concerned, session management relates to securing and managing multiple users' sessions against their request. It facilitates secure interactions between a user and a service/application and pertains to a sequence of requests and responses associated with a particular user. Session Management is one of the most critical aspects of Spring security as if sessions are not managed properly, the security of data will suffer. To control HTTP sessions, Spring security uses the following options:

* SessionManagementFilter.
* SessionAuthneticationStrategy

With these two, spring-security can manage the following security session options:

* Session timeouts (amount of time a user can remain inactive on a website before the site ends the session.)
* Concurrent sessions (the number of sessions that an authenticated user can have open at once).
* Session-fixation (an attack that permits an attacker to hijack a valid user session).

### What is JWT?

JWT (JSON Web Tokens) are tokens that are generated by a server upon user authentication in a web application and are then sent to the client (normally a browser). As a result, these tokens are sent on every HTTP request, allowing the server to verify or authenticate the user's identity. This method is used for authorizing transactions or requests between client and server. The use of JWT does not intend to hide data, but rather ensure its authenticity. JWTs are signed and encoded, instead of encrypted. A cryptographic algorithm is used to digitally sign JWTs in order to ensure that they cannot be altered after they are issued. Information contained in the token is signed by the server's private key in order to ensure integrity.